

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
13 September 2001 (13.09.2001)

PCT

(10) International Publication Number
WO 01/67240 A1

(51) International Patent Classification⁷: **G06F 9/44, 9/45, G07F 7/10** (FR). **RANGONI, Gabriel** [FR/FR]; 28, Boulevard de Lesseps, F-78000 Versailles (FR).

(21) International Application Number: **PCT/IB01/00343**

(74) Agent: **DEN BRABER, Gérard**; Schlumberger Systèmes, 50, avenue Jean Jaurès, Boîte postale 620 12, F-92542 Montrouge (FR).

(22) International Filing Date: **9 March 2001 (09.03.2001)**

(25) Filing Language:

English (81) Designated States (*national*): CN, JP, US.

(26) Publication Language:

English (84) Designated States (*regional*): European patent (DE, ES, FR, GB, IT, NL).

(30) Priority Data:

00/03144 10 March 2000 (10.03.2000) FR

Published:

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

(71) Applicant (*for all designated States except US*):

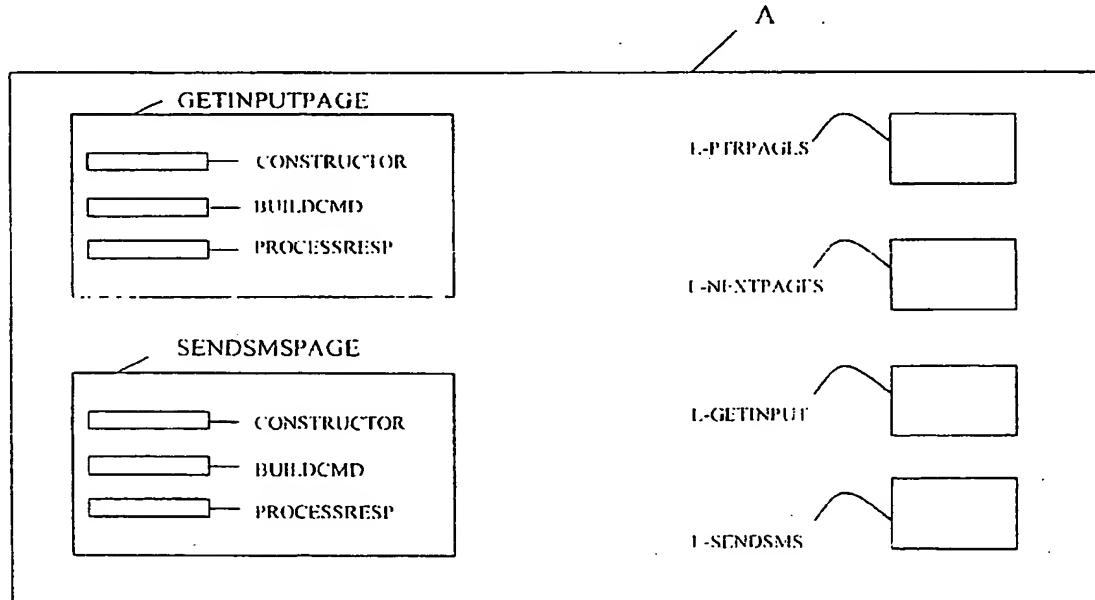
SCHLUMBERGER SYSTEMES [FR/FR]; 50, avenue Jean Jaurès, F-92120 Montrouge (FR).

(72) Inventors; and

(75) Inventors/Applicants (*for US only*): **IKBEL AMOR, Mohamed** [FR/FR]; 03, rue des Jonquilles, F-75014 Paris

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: AN INTEGRATED CIRCUIT DEVICE INCLUDING AN APPLICATION PROGRAM



WO 01/67240 A1

(57) **Abstract:** The invention relates to an integrated circuit device comprising a memory and at least one application program resident in said memory. The invention is characterized in that said application program comprises a list of application command pointers, a chaining list for such commands, a class associated with each command, and a configurable parameter list associated with each class, and also each class includes command parameterization means and search means for searching for a following command together with its parameters. The invention is particularly applicable to smart cards.

AN INTEGRATED CIRCUIT DEVICE INCLUDING AN APPLICATION
PROGRAM

FIELD OF THE INVENTION

5 The present invention relates to an integrated circuit device including a memory and at least one application program residing in said memory, said program using at least one application command. The invention also relates to a method of managing an application 10 program in such a device.

Said devices are constituted in particular by portable articles such as smart cards that contain application programs concerning the fields of health, mobile telephony, or indeed banking.

15

BACKGROUND OF THE INVENTION

In general, smart cards comprise a card body having an electronic module integrated therein, where the module conventionally comprises a control element (e.g. a 20 central processor unit or CPU) and a memory. Said memory has at least one application program using application commands that enable dialog to be established with a terminal, for example a mobile telephone in the field of 25 telephony. The application program can be loaded into the card via a communications network, and this applies in particular to application programs coded in an object-oriented language, e.g. JAVA (registered trademark). In general, an application program relates to a service offered by a card supplier to a client. For example, the 30 supplier can be a telecommunications operator and the client can be the user of a portable telephone in which the smart card is inserted.

In order to create such application programs, the state of the art proposes apparatuses that seek to 35 minimize the use of object components (classes and instances) and also to minimize hierarchization of classes, since such objects and such hierarchization take

up a great deal of memory space. This serves to reduce the size of the program code.

Although such devices make it possible to reduce code size in memory, this nevertheless remains a major problem in smart cards where memory is very limited. In addition, the time required to develop an application program remains high since application programs must be adapted to each particular service offered by the card supplier to a client. In addition, the code for an application program is difficult to reuse in any other service. Finally, if a minor modification is made in an application program, then it is necessary to reload the entire program in cards, which gives rise to a problem when cards have already been issued, i.e. when they have already been sold to clients.

SUMMARY OF THE INVENTION

It is an object of the present invention to propose an integrated circuit device comprising a memory and at least one application program resident in said memory, said program using at least one application command, and also to provide a method of managing and application program in such a device, making it possible firstly to reduce significantly the size of the code of an application program and secondly to reduce the time required to develop an application program by reusing a large portion of the code from another application program, and finally to be able to modify an application program after the device has been issued, without that making it necessary to reload said program into said device.

In a first aspect of the present invention, a solution to the technical problem posed is characterized in that firstly said application program has a list of application command pointers, a chaining list for said commands, a class associated with each command, and a list of configurable parameters associated with each

class, and secondly, each class has command parameterization means and search means for searching for a following command together with its parameters.

5 In a second aspect of the present invention, this solution is characterized in that the management method comprises the steps consisting in:

10 - creating in said application program a list of application command pointers, a chaining list for said commands, and a class associated with each application command;

- creating, in said application program, a list of configurable parameters associated with each class;

- parameterizing an application command;

- executing said command; and

15 - searching for the following command together with its parameters.

20 Thus, as described in greater detail below, the apparatus of the invention makes it possible by means of the list of pointers and the command chaining to avoid coding an entire chain of commands, thereby reducing the size of the code quite considerably. In addition, it is possible in the lists to modify the command chaining parameters directly without needing to modify and reload an entire application program. Finally, since the 25 characteristics such as the chaining and the parameters of the commands of a service associated with an application program are in lists and not in a class, a class is generic and can thus be reused in another application program, thereby also reducing development 30 time.

The following description made with reference to the accompanying drawings given by way of non-limiting example makes it easy to understand what the invention consists in and how it can be implemented.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a diagram of an integrated circuit device, in this case a smart card, including an application program.

5 Figure 2 is a diagram of the application program of the Figure 1 card.

Figure 3 gives a list of command pointers contained in the application program of the Figure 1 card.

10 Figure 4 gives a command chaining list contained in the application program of the Figure 1 card.

Figure 5 gives lists of parameters contained in the application program of the Figure 1 card.

Figure 6 is a diagram showing how an application program of Figure 1 is executed.

15

DETAILED DESCRIPTION

The present description of the invention relates by way of example to integrated circuit cards that are also known as "smart cards". The term "integrated circuit card" is used, for example, to mean a card in ISO format, a card in the so-called "plug-in" format such as a subscriber identity module (SIM), or indeed an electronic label.

25 Figure 1 shows an integrated circuit device 10, in this case a smart card. The card 10 has a command element 11 (e.g. a central processor unit or CPU), a memory 12, and a block 13 of contacts for making electrical connections with a terminal T, e.g. a connector in a card reader. The memory 12 contains an application program A. The memory 12 is preferably non-volatile and rewritable.

30 The smart card is generally supplied to a client by a service supplier, for example an operator managing a telecommunications network. Under such circumstances, the client is a user of the card and possesses a terminal known as a mobile telephone. The card is inserted in the mobile telephone. An application program A is preferably

associated with one or more services offered by the supplier. For example, there exists a service that enables a user to send messages over the operator's network by initially entering a password into the telephone.

The program A is shown in Figure 2. In this example, the program A uses a generic application command GETINPUT for text input and a generic application command SENDSMS for sending messages. Advantageously, each generic application command is associated with a class. Thus, a first class GETINPUTPAGE and a second class SENDSMSPAGE are respectively associated with each of the generic application commands of the example. In addition, and preferably, each generic application command is associated with one or more specific application commands. A generic application command preferably includes a set of parameters that are common to all associated specific commands, while a specific application command includes parameters that are specific thereto.

Program A also comprises:

- a list of generic command pointers L_PTRPAGES;
- a command chaining list L_NEXTPAGES; and
- a list of configurable parameters associated with each class, i.e. in this example a first list of text input parameters L_GETINPUT and a second list of message sending parameters L_SENDSMS.

The command chaining list L_NEXTPAGES has at least one reference relating to the list of command pointers L_PTRPAGES and at least one reference relating to a list of configurable parameters.

Each class has means for parameterizing commands, in this case an operational parameterizing command BUILDCMD, means for searching for a following command together with its parameters, in this case an operational search command PROCESSRESP, and command constructor means CONSTRUCTOR also known as a class constructor in object-

oriented language. Furthermore, and preferably, a class can have a location parameter OFFSETNEXTPAGES enabling the location of the command chain in the chaining list L_NEXTPAGES to be situated. The command constructor 5 means CONSTRUCTOR makes it possible to give a value to said parameter OFFSETNEXTPAGES. Each constructor is called during loading of the application program A in the smart card 10. A constructor makes it possible to create an instance of a class, i.e. to create an object of the 10 type of the associated class.

In order to enable the program A to execute, it is necessary to begin by creating the above-described lists and classes.

15 In a first step, in said application program A, the following are created: the list of application command pointers L_PTRPAGES, the chaining list for such commands L_NEXTPAGES, and the classes GETINPUTPAGE, SENDSMSPAGE which are associated respectively with the generic text input and send message commands.

20 As shown in Figure 3, the pointer list L_PTRPAGES preferably includes a first non-addressed index pointer PTR1, a second index pointer PTR2 pointing to an instance of the first class GETINPUTPAGE, and a third index pointer PTR3 pointing to an instance of the second class SENDSMSPAGE. The list does not necessarily have a 25 pointer for a single class instance, but could also have pointers for a plurality of instances of the same class.

30 In a second step, the following are created in said application program A: the lists of configurable parameters L_GETINPUT and L_SENDSMS associated with each class or generic command.

In a third step, a link is constructed between the command chaining list L_NEXTPAGES and the application commands.

35 To this end, the search means PROCESSRESP is created in each class, said means using the chaining list L_NEXTPAGES.

The command chaining list L_NEXTPAGES has chains of commands relating to the services offered by the telecommunications operator to the client, i.e. in the example of sending a message, the following chain:

5 - input a password, using a first specific text input command GETINPUT1;

 - input the text of the message to be sent, using a second specific text input command GETINPUT2; and

10 - send the message over the network using a send message command SENDSMS1.

To this end, and advantageously, the chaining list L_NEXTPAGES includes, for each specific command implying a following command OK:

15 - a reference PTR_OK to the following command to be executed in the list of command pointers L_PTRPAGES; and

 - a reference OFFSET_OK relating to the list of parameters associated with the referenced following command.

20 Advantageously, said list also includes, for each specific command implying a return command BACK:

 - a return command reference PTR_BACK for execution in the list of command pointers L_PTRPAGES; and

 - a reference OFFSET_BACK relating to the list of parameters associated with the reference return command.

25 When there is no return command or no following command, the references PTR_BACK and PTR_OK are preferably not addresses, the first index pointer PTR1 is allocated to them and the references OFFSET_BACK or OFFSET_OK are given a value zero.

30 Preferably, a reference is provided on one byte.

 Thus, in our example, and as shown in Figure 4, the following references will be obtained together with the following values:

35 1) for the first specific text input command GENTINPUT1,

OK = GETINPUT2

 - PTR_OK = PTR2

- OFFSET_OK = 02
BACK = 00
- PTR_BACK = PTR1
- OFFSET_BACK = 00
5 2) for the second specific text input command
GETINPUT2:

OK = SENDSMS1
- PTR_OK = PTR3
- OFFSET_OK = 01
10 BACK = GETINPUT1
- PTR_BACK = PTR2
- OFFSET_BACK = 01

The send message command SENDSMS1 is not in the list
15 L_NEXTPAGES since it does not imply any following command
or any return command. It will thus be understood that
the associated class SENDSMSPAGE does not have a location
parameter OFFSETNEXTPAGES.

Thereafter, and advantageously depending on the
location parameter OFFSETNEXTPAGES and the value of the
20 reference OFFSET in the list of parameters, and using the
operational search command PROCESSRESP, the appropriate
application command is found together with its parameters
in the associated parameter list.

In a fourth step a link is constructed between the
25 configurable parameter lists and the associated
application commands.

To this end, the parameterization means BUILDCMD is
created in each class, said means using an associated
parameter list. Thus, the parameterization means
30 BUILDCMD of the first class GETINPUTPAGE uses the list of
text input parameters L_GETINPUT, while the
parameterization means BUILDCMD of the second class
SENDSMSPAGE uses the send message parameter list
L_SENDSMS.

35 Preferably, each specific command is associated with
a group of parameters in the parameter list of the
associated class or the associated generic command.

Thus, in the example of Figure 5, the text input parameter list L_GETINPUT has two groups of parameters G1 and G2 corresponding respectively to the two specific text input commands GETINPUT1 and GETINPUT2. Each group 5 has four parameters MIN, MAX, TYPE, RFU with values coded on one byte each. The first and second parameters MIN and MAX represent the minimum and maximum length of the text to be input, the third parameter TYPE represents the type of text input, e.g. text made up of ASCII 10 characters, and the fourth parameter RFU is reserved for future use.

Thus, the password input must have a length of four characters, while the message to be sent must have a minimum length of one character and a maximum length of one hundred and seventy characters.

Similarly, in a non-limiting embodiment, the send message list L_SENDSMS has a single group of parameters G1 with a single parameter TAB indicating the number of the table from which all of the parameters of the 20 specific send message command SENDSMS1 are to be found. The table is the first table. It would have been easy to put the set of parameters for the second specific send message command in the list L_SENDSMS or indeed to put the set of parameters for a plurality of specific 25 commands in a single table and to reference the different groups of parameters in the list L_SENDSMS by means of a search reference associated with each group.

When the user sends a message by means of a mobile telephone, the application program is executed. As shown 30 in Figure 6, execution begins by the first text input command GETINPUT1. The location parameter OFFSETNEXTPAGES has already been initialized, in this case to zero.

Thereafter, the command is parameterized by means of 35 the operational parameterization command BUILDCMD of the first class GETINPUTPAGE. The list of parameters L_GETINPUT is used and the parameters are recovered by

using the reference OFFSET_OK as initialized previously to the value 01.

The calculation performed is then as follows:

4 * (OFFSET_OK - 1) to recover the first parameter,
5 4 * (OFFSET_OK - 1) + 1 for the second parameter, etc.

The command is sent to the terminal T and executed. A message is displayed on the terminal T prompting the user to enter the password. The user enters the password 1234 and presses the BOK key of the telephone. The 10 terminal sends a response to the card 10 together with the user's password, the response corresponding to the result of executing the command. The response is preferably encoded on one byte. In this case, the response has the value zero. In a different example, if 15 the user had pressed the BBACK key, then the response would have had hexadecimal value 11, etc.

The following command together with its parameters are searched for by the means of the operational search command PROCESSRESP of the first class GETINPUTPAGE.

This command uses the chaining list L_NEXTPAGES. Also because of the location parameter OFFSETNEXTPAGES, the chaining of a specific application command is searched for, in this case the first specific application command GETINPUT1, in the chaining list L_NEXTPAGES.

The calculation performed is then as follows:

OFFSETNEXTPAGES + 4 * (OFFSET_OK - 1) + 2 for the following command, or else OFFSETNEXTPAGE + 4 * (OFFSET_BAC - 1) for the return command.

The following command is the second specific text 30 input command GETINPUT2 and the reference OFFSET_OK of the associated parameter list is 02. The position found is in the associated list of parameters L_GETINPUT at the reference OFFSET_OK. The command pointed to by the second index pointer PTR2 is called. The appropriate 35 generic command of the associated class GETINPUTPAGE is thus found.

The second specific command GETINPUT2 is prepared and parameterized by means of the operational parameterization command BUILDCMD of the associated class. This serves to recover the parameters from the 5 associated list L_GETINPUT at the reference OFFSET_OK whose value is 02. The second command GETINPUT2 is sent to the terminal T. It is executed. A message is displayed on the terminal T prompting the user to enter the message that is to be sent. The user enters the text 10 of the message MSG and presses on the BOK button.

A search is then made for the following command together with its parameters in the chaining list L_NEXTPAGES.

The following command is a specific send message 15 command SENDSMS1 and the reference OFFSET_OK of the associated parameter list has the value 01. The search leads to the associated parameter list L_SENDSMS at its reference OFFSET_OK. The command pointed to by the second index pointer PTR3 is called. This thus finds the 20 appropriate generic command in the associated class SENDSMSPAGE.

The specific command SENDSMS1 is prepared and parameterized by means of the operational parameterization command BUILDCMD of the associated 25 class. The parameters are thus recovered from the associated list L_SENDSMS at the reference OFFSET_OK of value 10 and consequently from the first table TAB. The specific command SENDSMS1 is sent to the terminal T. It is executed. The consequence of execution is to send the 30 message over the telecommunications network.

As shown above, an application program A is preferably associated with one or more services offered by the supplier. To this end, the user can select a service from a plurality of services by means of a menu 35 on a mobile telephone. For example, the user can select between the first send message service as described above

and a service for directly sending a message that is preprogrammed in the card 10.

Selection is executed by means of a generic selection application command SELECTITEM. Thereafter, an 5 associated class SELECTITEMPAGE is created comprising parameterization means BUILDCMD and command search means PROCESSRESP. In this case, the search means PROCESSRESP is similar to those described above. Since the selection command is generic, the parameterization means comprises 10 solely means for preparing parameters in a buffer that is subsequently used while executing the command, and the selection command class does not have a list of parameters. It will be observed that this preparation means is included in each class and preferably in each 15 parameterization means BUILDCMD.

The chaining list L_NEXTPAGES has the following references and values:

For the specific selection command SELECTITEM1,
BACK = 00
20 - PTR_BACK = PTR1
- OFFSET_BACK = 00
SERV1 = GETINPUT1
- PTR_OK = PTR2
- OFFSET_OK = 01
25 SERV2 = SENDSMS2
- PTR_OK = PTR3
- OFFSET_OK = 02

In this example, said references are included initially in the list prior to the references of the 30 other classes.

Thus, if the user selects the first service SERV1, the command to be executed is the first text input command GETINPUT1, and so on as described above. If the user selects the second service SERV2, the command to be 35 executed is a second send message command SENDSMS2 whose parameters are, for example, recovered from a second table TAB. Naturally, the parameter list L_SENDSMS is

updated accordingly with a second group G2. If the user presses on a return key BBACK, then the user leaves the application program.

When a user selects a service, the terminal sends a response comprising an associated service number to the card 10. Thus, the first service SERV1 has the associated number 01, and the second service SERV2 has the associated number 02. The number associated with the "service" of the button BBACK is 0.

The calculation performed in the search command PROCESSRESP is then as follows: OFFSETNEXTPAGES + 2*(service number) for the following command or else OFFSETNEXTPAGES for the return command, with the location parameter OFFSETNEXTPAGES at zero.

Whereas for text input commands and message sending commands the calculation performed is as follows: OFFSETNEXTPAGES + 4*(OFFSET_OK - 1) + 2 for the following command, otherwise OFFSETNEXTPAGES + 4*(OFFSET_BACK - 1) for the return command, with the location parameter OFFSETNEXTPAGES having a value of six.

It will thus be understood that the subject matter of the invention has the advantage of considerably reducing code size by means of the above lists, clear separation between generic elements and elements that are more specific to a service. This separation makes it easier to reuse generic elements such as classes in other application programs and consequently shortens development time. In addition, if one or more services are to be modified, this is easily done, even with cards that have already been issued, by making modifications to the lists. By way of example, the modifications can be performed by downloading data over a radio telecommunications network using the known short message technique. Finally, the subject matter of the invention as described above makes it possible to manage various kinds of response from the terminal T sent to the card 10, with the responses corresponding to different results

when executing a command. In the examples above, two kinds of response are handled, when the user presses on the BOK key or when the user presses on the BBACK key of the mobile telephone. It is also possible to handle 5 responses due to errors such as non-execution of a command, no response by the user, At this moment, from the above, there will always be a following command associated in the chaining list L_NEXTPAGES with each response from the terminal to be processed. It will thus 10 be understood that in each class, there is no longer any need for coding to handle various errors, thereby considerably reducing code size.

CLAIMS

1/ An integrated circuit device comprising a memory and at least one application program (A) resident in said memory, said program using at least one application command, the device being characterized in that firstly said application program (A) has a list of application command pointers (L_PTRPAGES), a chaining list for said commands (L_NEXTPAGES), a class associated with each command, and a list of configurable parameters associated with each class, and secondly, each class has command parameterization means (BUILDCMD) and search means for searching for a following command together with its parameters (PROCESSRESP).

15 2/ A device according to claim 1, characterized in that the command chaining list (L_NEXTPAGES) has at least one reference relating to the list of command pointers (L_PTRPAGES) and at least one reference relating to a list of configurable parameters.

20 3/ A method of managing an application program of an integrated circuit device comprising a memory and at least one application program (A) residing in said memory, said program (A) using at least one application command, the method being characterized in that said method comprises the steps consisting in:

- creating in said application program a list of application command pointers (L_PTRPAGES), a chaining list for said commands (L_NEXTPAGES), and a class associated with each application command;
- creating, in said application program, a list of configurable parameters associated with each class;
- parameterizing an application command;
- executing said command; and

35 - searching for the following command together with its parameters.

4/ A method according to claim 3, characterized in that the command chaining list (L_NEXTPAGES) has at least one reference relating to the list of command pointers (L_PTRPAGES) and at least one reference relating to a list of configurable parameters.

5

1/3

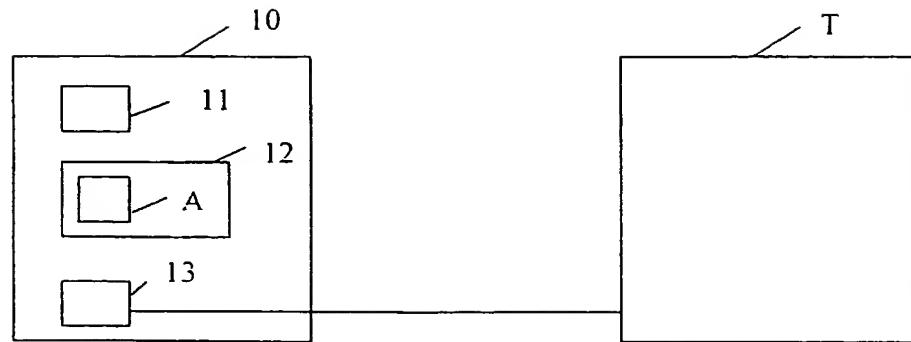


FIG. 1

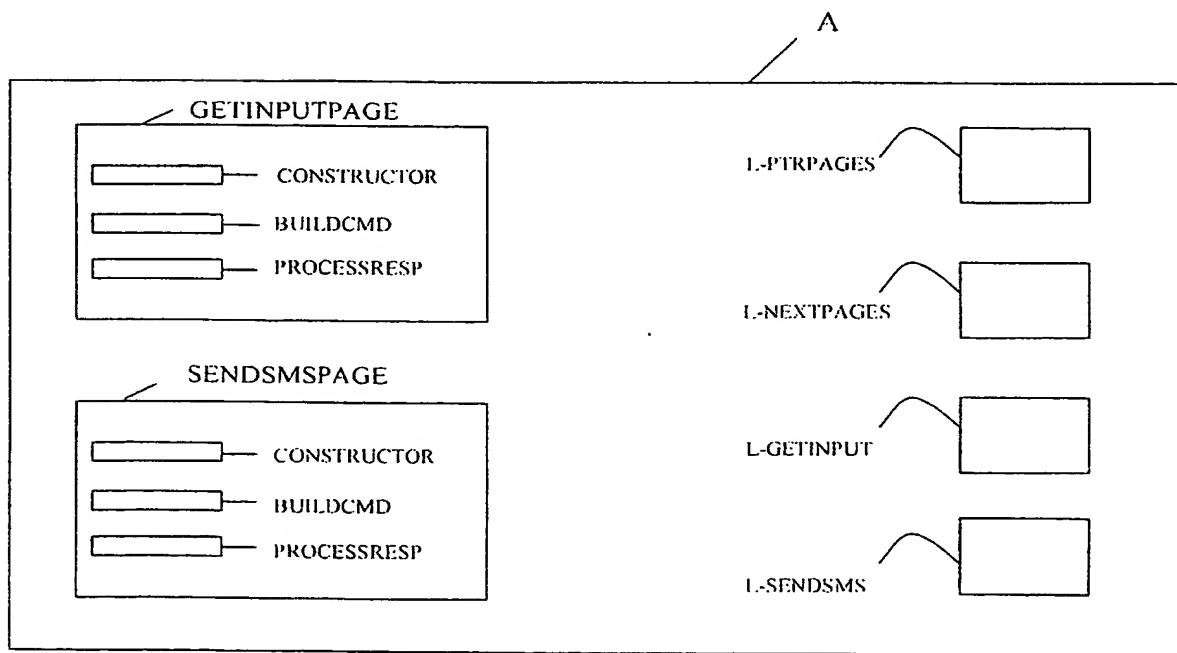


FIG. 2

2/3

L-PTRPAGES

L-PTRPAGES [PTR1] = NULL
 L-PTRPAGES [PTR2] = new GETINPUTPAGE
 L-PTRPAGES [PTR3] = new SENDSMSPAGE

FIG. 3

L-NEXTPAGES

GETINPUT1		GETINPUT2	
BACK	OK	BACK	OK
PTR-BACK	OFFSET-BACK	PTR-OK	OFFSET-OK
PTR1	00	PTR2	02
		PTR2	01
		PTR3	01

FIG. 4

3/3

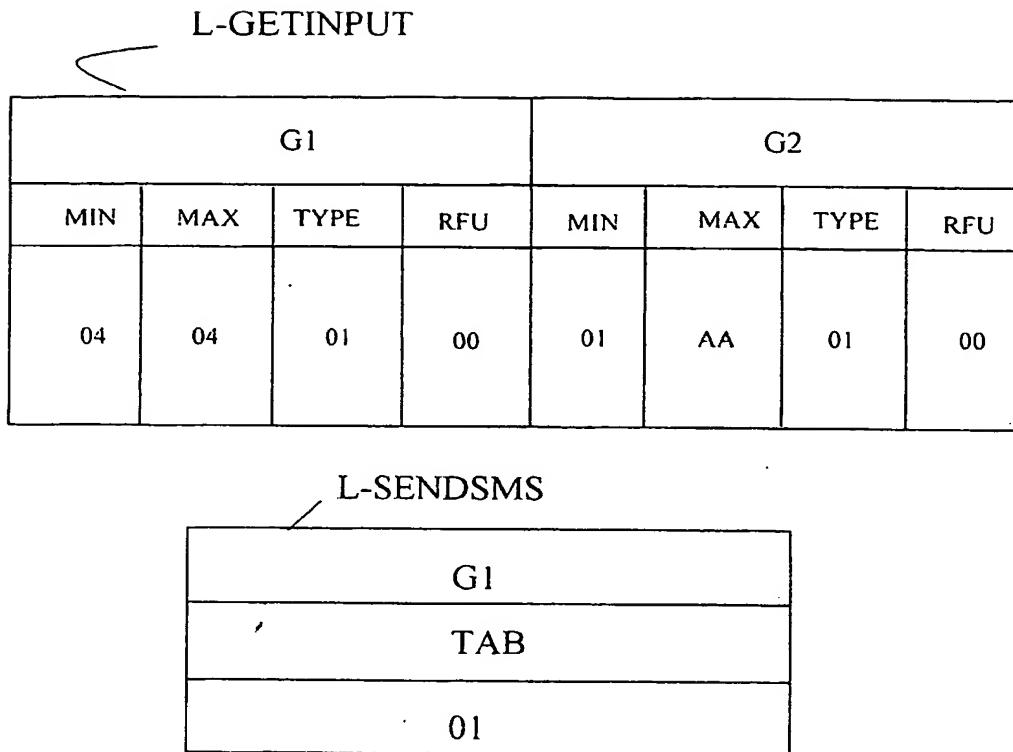


FIG. 5

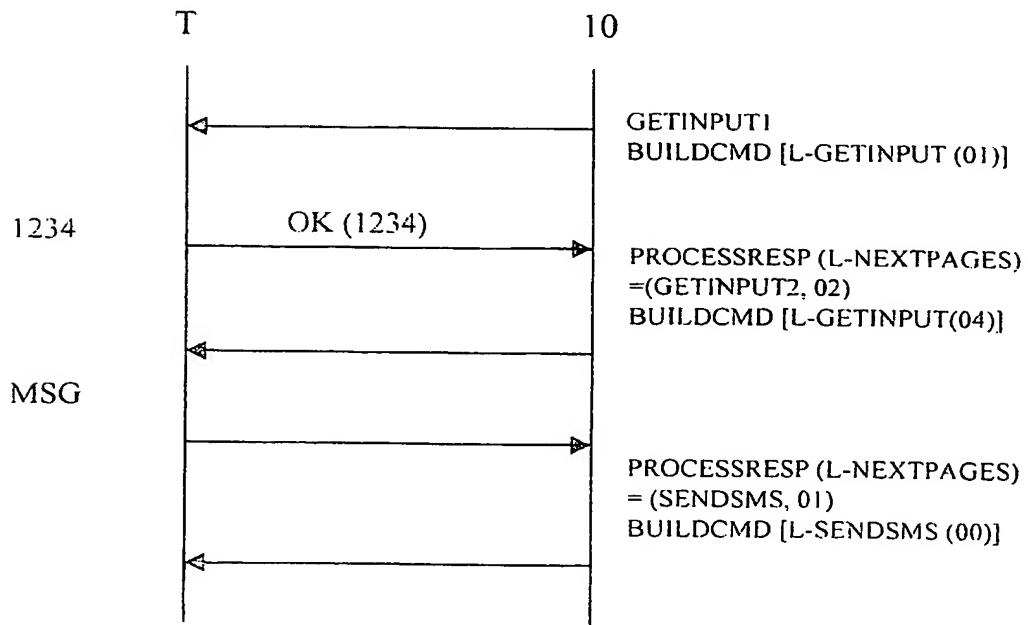


FIG. 6

INTERNATIONAL SEARCH REPORT

International Application No
PCT/IB 01/00343A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F9/44 G06F9/45 G07F7/10

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F G07F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Character of document with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	FR 2 757 970 A (GEMPLUS CARD INT) 3 July 1998 (1998-07-03) claims 1-3,6 ---	1-4
A	EP 0 626 664 A (GEMPLUS CARD INT) 30 November 1994 (1994-11-30) abstract column 1, line 32 -column 2, line 27; claims 1,2 ---	1-4 -/-

 Further documents are listed in the continuation of box C. Patent family members are listed in annex.

* Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *&* document member of the same patent family

Date of the actual completion of the international search 26 June 2001	Date of mailing of the international search report 06/07/2001
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Paraf, E

INTERNATIONAL SEARCH REPORT

International Application No PCT/IB 01/00343	
---	--

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	BIGET P ET AL: "How smart cards can benefit from object-oriented technologies" FUTURE GENERATIONS COMPUTER SYSTEMS, NL, ELSEVIER SCIENCE PUBLISHERS. AMSTERDAM, vol. 13, no. 1, 1 July 1997 (1997-07-01), pages 75-90, XP004081711 ISSN: 0167-739X page 83, column 2, paragraph 2 -page 84, column 1, paragraph 2 page 85, column 1, paragraph 3 ----	1-4
A	DEBAERE E H: "A language coprocessor for the interpretation of threaded code" MICROPROCESSING AND MICROPROGRAMMING, NL, ELSEVIER SCIENCE PUBLISHERS, BV., AMSTERDAM, vol. 21, no. 1/05, August 1987 (1987-08), pages 593-602, XP002114520 ISSN: 0165-6074 page 594, column 1, paragraph 1 -page 595, column 2, paragraph 4 figure 2 ----	1-4
P,A	FR 2 786 901 A (SCHLUMBERGER SYSTEMES) 9 June 2000 (2000-06-09) abstract; claims 1,3-6; figures 2-4 ----	1-4
P,A	FR 2 785 695 A (BULL CP8) 12 May 2000 (2000-05-12) page 4, line 3 -page 5, line 6 claims 1,3,4 ----	1-4

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/IB 01/00343

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
FR 2757970	A 03-07-1998	AU 733631	B	17-05-2001
		AU 5769598	A	31-07-1998
		BR 9713630	A	11-04-2000
		CN 1247609	A	15-03-2000
		EP 1012710	A	28-06-2000
		WO 9829803	A	09-07-1998
		JP 2001500655	T	16-01-2001
EP 0626664	A 30-11-1994	FR 2704704	A	04-11-1994
		DE 69400549	D	24-10-1996
		DE 69400549	T	30-01-1997
		ES 2092867	T	01-12-1996
		JP 7073281	A	17-03-1995
		SG 48143	A	17-04-1998
		US 5687398	A	11-11-1997
		US 5896507	A	20-04-1999
FR 2786901	A 09-06-2000	WO 0034927	A	15-06-2000
FR 2785695	A 12-05-2000	BR 9906784	A	17-10-2000
		CN 1292114	T	18-04-2001
		EP 1046105	A	25-10-2000
		WO 0028416	A	18-05-2000

THIS PAGE BLANK (USPTO)